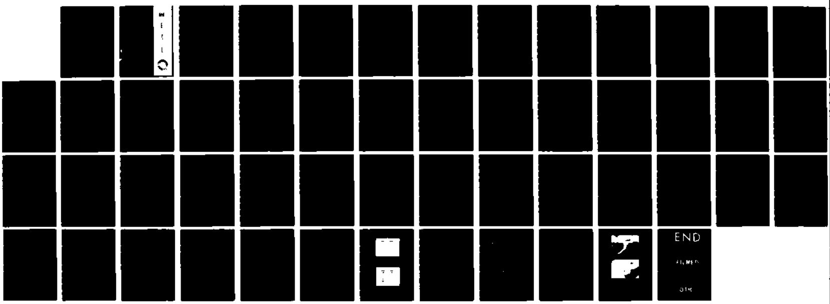
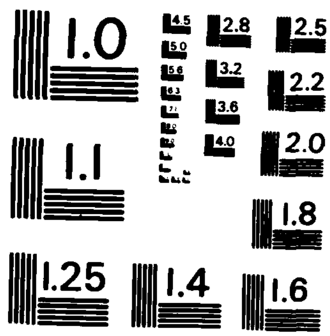
AD-A160 642 A VISUAL NAVIGATION SYSTEM FOR AUTONOMOUS LAND VEHICLES 1/1
(U) MARYLAND UNIV COLLEGE PARK CENTER FOR AUTOMATION
RESEARCH A M WAXMAN ET AL. OCT 85 CAR-TR-139 ETL-0408
UNCLASSIFIED DACA76-84-C-0004 F/G 17/7 NL




1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

2

ETL-0408

AD-A160 642

A visual navigation system for autonomous land vehicles

Allen M. Waxman
Jacqueline LeMoigne
Larry S. Davis
Eli Liang
Tharakesh Siddalingaiah

Center for Automation Research
University of Maryland
College Park, Maryland 20742

October 1985

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.

Prepared for
U.S. ARMY CORPS OF ENGINEERS
ENGINEER TOPOGRAPHIC LABORATORIES
FORT BELVOIR, VIRGINIA 22060-5546 85

E T L

U.S. ARMY ENGINEER TOPOGRAPHIC LABORATORIES
ETL

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ETL–0408	2. GOVT ACCESSION NO. AD.A 16	3. RECIPIENT'S CATALOG NUMBER 642
4. TITLE (and Subtitle) A VISUAL NAVIGATION SYSTEM FOR AUTONOMOUS LAND VEHICLES		5. TYPE OF REPORT & PERIOD COVERED Technical Report July 1984 - July 1985
		6. PERFORMING ORG. REPORT NUMBER CAR-TR-139, CS-TR-1537
7. AUTHOR(s) Allen M. Waxman, Eli Liang Jacqueline LeMoigne, Tharakesh Siddalingaiah Larry S. Davis		8. CONTRACT OR GRANT NUMBER(s) DACA76–84–C–0004
9. PERFORMING ORGANIZATION NAME AND ADDRESS Center for Automation Research University of Maryland College Park, Maryland 20742		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS U.S. Army Engineer Topographic Laboratories Fort Belvoir, Virginia 22060-5546		12. REPORT DATE October 1985
		13. NUMBER OF PAGES 49
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer vision, Road following,
Image processing, Rule-based reasoning,
Autonomous navigation.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Computer Vision Laboratory at the University of Maryland has developed a modular system architecture to support visual navigation by an autonomous land vehicle. The system consists of vision modules performing image processing, 3-D shape recovery, and rule-based reasoning, as well as modules for planning, navigating, and piloting.

DD FORM 1473, 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

PREFACE

This document was prepared under contract number DACA76–84–C–0004 for the U.S. Army Engineer Topographic Laboratories, Fort Belvoir, Virginia by the Center for Automation Research, University of Maryland, College Park, Maryland. The Contracting Officer's Representative was Ms. Rosalene M. Holecheck.

CONTENTS

1. INTRODUCTION

This work concerns the development of a general framework for visual navigation over terrain and roadways. The objective is to endow a mobile robot with the intelligence required to sense and perceive that part of its surroundings necessary to support navigational tasks. To this end, we have designed and implemented a modular system which is currently able to drive a camera over a scale model road network strictly under the control of vision (i.e. no a priori map is provided). The vehicle maintains continuous motion by alternatively "looking ahead" and then "driving blind" for a short distance before taking another view. While moving blindly, the accepted monocular image is processed to extract features which are then interpreted in three dimensions by a combination of "shape from contour" and rule-based reasoning. This provides the data required to form an object-centered representation in the form of a local map. This map is used both for navigation and for focusing attention on selected parts of the visual field as the vehicle continues to move, accepting new images for processing. Though our domain of application has been the visual navigation of roadways, we have attempted to derive some useful principles relevant to visual navigation systems in general. Our preliminary efforts on this work have been reported in [1].

A variety of design philosophies for mobile robot systems have been discussed in the literature [2-7], and our approach has some features in common with them. The guiding principles that have emerged from our research touch on three areas of design. The first concerns the modularity of the system architecture, in which a distinction is made between computational capabilities within modules, the nature of the representations that enter and leave these modules, and the flow of control through the system. The second deals with the nature of the focus of visual attention in the system. This leads to significant computational savings and greater vehicle speeds. The third is a decomposition of general navigation tasks into three categories which support navigation on three different scales.

The construction of a modular system architecture is attractive for a variety of reasons. Individual modules can possess well defined responsibilities and consist of several related computational capabilities. The behavior that the vehicle itself exhibits then reflects the "flow of control" through the modules of the system. Though our current system is only able to navigate on simple roadways, we have put in place many of the computational capabilities that are required for more complex behavior. This decomposition into modules also points to the importance of the interfaces that link modules, for it is across these interfaces that intermediate representations of the processed data travel as well as the control commands which govern the choice of computations within modules. Modularity also opens the door to distributed parallel processing, with various modules possibly running on dedicated hardware. For example, some of our image processing routines, developed on a VAX 11/785, have been adapted to run on a VICOM image processor [8]. We will describe the modules which comprise our system, their current and planned capabilities for the future, and the flow of control which supports road following.

We have found it useful to distinguish between two different modes of visual processing, and our system can switch between these modes when necessary. Generally, the system begins a task in the *bootstrap mode* which requires processing an entire scene, picking out the objects of interest such as roads or landmarks. Sometimes this mode of processing can be avoided if the system is provided with detailed data concerning the location of such objects in a map, say, and the precise location of the vehicle in that map. We prefer to endow our system with the ability to operate independent of such information since it is not always available. Once objects of interest are identified, the system switches to a prediction-verification mode called *feed-forward* in which the location of an object as seen from a new vehicle position is estimated, thus focusing attention on a small part of the visual field. This feed-forward capability emerges from the interaction between vehicle dead-reckoning, 3-D world modeling and vision. It has been particularly useful for road following, leading to a computational saving of a factor of ten. However, it is a useful principle in general, and will be applicable to obstacle avoidance as well.

Navigational tasks have been categorized into three types (or levels) acting on different scales, and all three contribute to the general act of path planning for the vehicle. Also, all three types of tasks require their own visual and computational capabilities with supporting representations. These levels of navigation correspond to our decomposition and representation of space through which the vehicle must navigate. We denote these three levels as *long range, intermediate* *range, and short range navigation.* Each deals with space on a different scale and for a different purpose. Thus, long range navigation deals with the decomposition of space into a set of contiguous regions, each region possessing a certain uniformity of characteristics such as visibility of landmarks and traversibility. Intermediate range navigation is responsible for selecting corridors of free space between regions. Finally, short range navigation must determine tracks of safe passage through a corridor of free space, avoiding obstacles that are detected during the traverse. Related ideas in spatial reasoning have been described in [9-11].

We have implemented our system as a set of concurrent modules running on a time-sharing system (Berkeley UNIX 4.2) with the modules communicating through UNIX system "sockets." The visual navigation system is used to drive a robot arm carrying a small CCD camera (our vehicle) over a 1:96 scale model of a road network. This simulation has provided us with an adequate and flexible environment in which to develop and test further navigational capabilities for autonomous vehicles.

In Section 2 we describe the decomposition of navigation into three levels of path planning at different scales. Section 3 details our system architecture, its modular decomposition into functional capabilities and the flow of control which supports road following behavior. The workings of the individual modules are then dealt with in Section 4. The robot arm vehicle simulator is discussed in Section 5. Concluding remarks and future directions follow in Section 6.

2. MULTI-SCALE NAVIGATION

Typically, it is the responsibility of the Navigator Module to determine the current position of the vehicle, and plan a path through the environment from the current position to the location of the goal (as specified by the Planner Module). In order to perform its functions, the Navigator must have available to it a variety of computational capabilities as well as visual resources (provided by the Vision Modules within the system).

We have found it useful to decompose this problem into three distinct "levels of navigation" acting on different spatial scales: *long, intermediate and short range navigation.* The paths generated are then in the form of a "sequence of regions" (a long range result), a directional heading or "corridor of free space" connecting the current region with the next region in the sequence (an intermediate range result), and finally a "track of safe passage" through the selected corridor while avoiding obstacles (a short range result).

2.1 Long Range Navigation

Long range navigation concerns the decomposition of the environment into regions which share common properties such as uniform visibility of landmarks and navigability of the terrain. On this scale, path planning amounts to establishing a sequence of regions through which the vehicle must move. Information associated with landmarks and their locations, as well as traversibility of terrain, must be made available to the system a priori; it may be obtained by aerial or ground reconnaissance. This kind of information can reside in one or more databases within the system. For example, topography data may be associated directly with the Navigator Module, whereas visual models of landmarks should reside in the knowledge base of the vision system. The location of known landmarks should be correlated with the terrain maps. In this way, gross positioning of the vehicle can be determined by visually sighting (and recognizing) a set of landmarks in a variety of directions (cf. [12]). The uniform visibility of landmarks in a given region insures that a sub-set of the same landmarks can be used throughout that region.

Given terrain (or road) traversibility data and landmark visibility data, the Navigator must be able to partition the environment into regions of uniform traversibility/navigability. The regions can then be assigned to nodes of a graph (or leaves of a quadtree), with distance between centroids encoded along arcs connecting traversible, neighboring regions. Path planning at this level of resolution consists of establishing a "sequence of regions" which connect the "start" and "goal" regions. Thus, the Navigator must possess the computational capabilities to construct such graphs and create region sequences. Figure 1 shows an hypothetical decomposition of such a data base into regions, and two possible region sequences which join the start and goal regions. Related ideas on decomposing map data into meaningful regions have been described in [3,4,9,10,11]. Path planning through regions represented as leaves of a quadtree has been discussed in [13].

2.2 Intermediate Range Navigation

Intermediate range navigation is invoked in order to select a corridor of free space through which the vehicle will next travel. This corridor should lead from the current position toward the next region of the chosen sequence of regions. It should be chosen so as to be free of known obstacles whose locations are also correlated with the terrain map. Corridors of free space may be known to the system a priori, such as from (road) maps, or they may need to be established using vision. In either case, visual resources are required in order to find the corridor in the visual field of the vehicle and so establish its heading.

Intermediate range navigation puts the greatest demands on the vision capabilities of the system. In a sense, establishing a corridor of free space through a local environment requires an ability to do general scene analysis. However, it is not really necessary to completely label a scene (and model it in 3-D) in order to extract this corridor. It is only necessary to recognize that portion of the scene constituting such a corridor and to ignore the rest. Navigating roadways is a typical example of intermediate range navigation, and so we have come to realize that extracting only the road portion of a scene is what is relevant. This approach to "corridor extraction" allows us to focus on generic attributes of corridors, without worrying about understanding the entire scene. In general, 3-D vision (shape from stereo, motion, contour, etc.) or active ranging is necessary to support the extraction of corridors of free space. It is also necessary to represent these corridors in a 3-D world model. Much of our work on navigating roadways is relevant here, and will be discussed below.

2.3 Short Range Navigation

Short range navigation is responsible for selecting the actual path to be traversed through the established corridor of free space. It should be a "track of safe passage" avoiding obstacles as they are detected by the visual resources of the system. Short range navigation also involves updating position with respect to a local map, and modifying the contents of this map provided either a priori or by vision. Thus, it resembles a kind of "short term memory" of the system.

In order to achieve a short range navigation capability, the Navigator must be provided with (or have extracted) the corridor of free space through which a detailed path will be planned. This corridor, and the obstacles detected within it could be represented in the form of a dynamic map. This map, which implies an encoding of positional information, should also allow for other attributes to be associated with objects, such as their names, velocities (if moving) or time to contact. In order to frequently update position in the local map, conventional navigation techniques such as dead reckoning (using wheel shaft encoders and perhaps gyroscopes) should be exploited. Visual resources must support obstacle detection and localization (though not necessarily "recognition"). And of course, the Navigator must be able to plan a specific path which avoids the obstacles while keeping the vehicle within the corridor of free space.

2.4 Navigation Examples

By the nature of our decomposition of navigation, it is clear that certain levels of navigati ,n are utilized more frequently than others. Thus, region decomposition may take place only at the start of a mission with positional updates via landmark detection occurring periodically. Establishing a corridor of free space requires significant computational resources if little or no focus of attention is provided (cf. bootstrap mode below); however, once established, it may be quite simple to propagate this corridor out ahead of the vehicle (cf. feed-forward mode below). The selection of tracks of safe passage will be required most often in cluttered environments and has received much attention by researchers in the past [2,5].

Three examples of navigation make clear our decomposition into three distinct levels. The first is a hiker (person or robot) making his way through the woods. Given a crude map of the terrain and landmarks in the environment, the hiker chooses the general direction he will take, from one region to another. Every now and then he stops, looks around (toward high ground) to sight landmarks, and thereby establishes a rough position estimate. These acts comprise long range navigation. Looking straight ahead, the hiker searches for a path or clearing through the woods in order to select a heading through a corridor of free space: intermediate range navigation. And most often, the hiker looks down around his feet in order to avoid obstacles along his path: a short range navigation task. A second example is a vehicle navigating an extensive road network, such as between cities. In this case, long range navigation amounts to selecting a sequence of highways connecting the regions in which the start and goal are located. Road signs inform the driver of his approximate location, thus playing the role of landmarks. Staying within the confines of any piece of roadway (or lane), essentially road following, is the intermediate range navigation task. Avoiding other cars and obstacles then comprises short range navigation. The third example concerns a mobile robot navigating the hallways of a building. Given a floor plan of the building, regions of uniformity take on a semantic meaning (e.g. room, hallway, alcove) [11]. Signs in hallways, or the layout of the walls themselves (such as corners), can play the role of landmarks here. Corridors of free space are what buildings are generally made of, though traversing a large room also requires this intermediate range navigation. Short range navigation comes into play for avoiding stationary and moving obstacles such as desks, chairs, people and other robots. It is the second example which has concerned us the most. Currently, our system's capabilities lie in the realm of intermediate range navigation for purposes of following roadways using vision. We will soon be adding a short range obstacle avoidance ability.

The decomposition of the general path planning task, as described here, should be contrasted with an alternative philosophy described by Crowley [7]. He prefers to establish a priori locations in a map, and then plan routes as sequences of these pre-learned positions. An obstacle avoidance ability is still required in case a planned route is blocked.

3. SYSTEM ARCHITECTURE

We have developed and implemented the modular system architecture shown in Figure 2. There are a variety of reasons for constructing a modular system, some of which were mentioned in the Introduction. Perhaps the most important point to make, however, is that in a modular system such as ours, there is a clear distinction between the *computational capabilities* that reside within the individual modules and the *flow of control* over the interfaces between the modules which allows the vehicle to exhibit some behavior. Adding or modifying computational capabilities alone does not imply that the vehicle can necessarily do something "new." A genuinely new capability corresponds to a new flow of control through the system, accessing the required computational capabilities at the appropriate times. This distinction allows one to concentrate separately on issues of computation and system behavior.

The contents of any one module is motivated by the similarity of the input and output representations of the data passing through that module. Thus, the procedures within a module act to transform the data between various intermediate representations. Transformations of similar type usually require similar computational resources, thus suggesting that individual modules be provided adequate computing power which may differ between modules. Hence, a modular architecture opens the way to concurrent processing on distributed hardware. True parallelism is achieved once the timing and synchronization of the system has been resolved. However, we do not believe that perception and cognition are "real-time" acts, though they are "time sensitive." Thus, the system cannot be perfectly synchronized at all times and so it must have a means of accommodating the asynchronous passage of data and control over the interfaces between modules.

3.1 Bootstrap vs. Feed-Forward

The architecture shown in Figure 2 consists of a vision system along with modules for Planning, Navigating and Piloting. The vision system is decomposed into modules which support low, intermediate and high-level vision. The representations which travel across the interfaces of the vision system reflect the various stages of visual processing as described by Marr [14]. This will become more clear as we describe the flow of control and data through the system in support of road following.

The vision system operates in two modes, *bootstrap and feed-forward.* This distinction is possible because the system can exploit a "focus of attention" on a portion of the visual field deemed important. The bootstrap mode is employed when the vision system must establish its first view of an object, i.e., it must find it in the visual field. This often requires processing the entire image. (If vehicle position is accurately known and detailed map data is provided, the system can bypass the bootstrap mode and use the map to focus attention.) Once an object is localized, the vision system can predict (to some accuracy) the location of that object in the following view and so restrict future processing to a small portion of the visual field. The system is then operating in the feed-forward mode. This kind of processing, in which a focus of attention is maintained, is particularly important when computational resources are limited, which is really always the case. For the task of road following, the bootstrap process, in which the road is identified in the field of view requires about 90 seconds of CPU time on a VAX 11/785. The feed-forward process, in which the location of the near part of the road is *predicted* to fall within a $5^o \times 5^o$ window, detection within that window is achieved, and the road is then tracked to a distance of some 60 meters, requires only 9 seconds of CPU time on a VAX 11/785. Thus, a factor of ten is gained in computational resources. In fact, it is via the feed-forward mode of operation that the vehicle can achieve continuous movement over an obstacle free road. By accepting an image for feed-forward processing, the vehicle can move "blindly" for a distance of about 10 meters at a speed of about 3 km/hr while deriving a new 3-D model of the road from the image. During this travel, the vehicle is dead-reckoning its position relative to the previously derived 3-D model. Depending on the accuracy of the dead-reckoning system, one can extend the distance of blind travel and so achieve greater vehicle speeds. However, this will not be appropriate if obstacles can move into the vehicle's path during the blind travel time. Alternatively, fast, dedicated hardware can be utilized to speed up the computations while keeping the blind travel distance to a minimum. In this spirit, an implementation of the image processing for the feed-forward mode on a VICOM image processing system cut the computation time to 4 seconds [8].

We shall describe the system architecture in Figure 2 by first explaining the responsibilities of the individual modules in terms of how they transform the data or representations traveling over the interfaces. Then we describe the flow of control through the system in support of the road following task. Further details of the computational capabilities within the modules are given in the next section.

3.2 Module Responsibilities

The Vision Modules in our system (cf. Fig. 2) support low-level (Image Processing), intermediate-level (Geometry) and high-level (Knowledge Based) vision. These modules are coordinated by a Vision Executive Module which also has access to 3-D Representation, Scene Prediction and Sensor Control modules. The Vision Executive also supports interfaces to the Planner and Navigator Modules. The Pilot Module communicates only with the Navigator. Theses modules have certain functional capabilities which reflect their individual responsibilities. These capabilities are listed inside each module in the diagram, and are described in detail in the next subsection.

The Vision System as a whole is responsible for perceiving objects of interest (e.g., roads and landmarks) and representing them in an "object centered" reference frame. The "Image Processing Module" is responsible for extracting symbolic representations from the individual images. These image domain symbolics correspond to significant events in the signal data; they are general features which describe images (e.g., edges, lines, blobs). The transformation from TV signals to symbols represents an enormous reduction in data. This module constructs a representation akin to the "raw primal sketch" of Marr [14]. Extraction of symbols can be performed either on the entire image, or within a specified window. The module we call "Visual Knowledge Base" has several responsibilities. Given the image domain symbolics extracted by the Image Processing Module, the Knowledge Based module tries to establish significant groupings of these symbols (e.g., pencils of lines). These groupings are global, corresponding to spatial organizations over large parts of the image, in contrast to the symbols themselves which are typically local groupings of events. This grouping process will then discard those symbols which are not found to belong to any group. This results in a representation similar to the "full primal sketch" of Marr [14]. The Visual Knowledge Base is also responsible for establishing meaningful groupings from 3-D representations provided by the "Geometry Module". Given the 3-D data, the Knowledge based module tries to recognize specific kinds of objects (e.g., roads), and so label important parts of the scene. The "Geometry Module" is responsible for 3-D shape recovery, converting the grouped symbolics (obtained earlier) into surface patches described in a viewer centered reference frame, as in the "2.5-D sketch" of Marr [14]. The "Vision Executive" is the heart of the Vision System; it maintains the "flow of control" through this part of the system, trying to meet the "attentive goals" (such as *find road* or *find landmark* or *find obstacles*) provided by the Planner and Navigator. The Vision Executive passes the intermediate representations among the modules of the Vision System, buffering them when necessary in order to support parallel processing of multiple images (residing in different modules at different stages of processing). It is this Executive which triggers the mode of operation (bootstrap or feed-forward) as well as selecting the relevant "knowledge packets" which reside in the Visual

Knowledge Base. The Vision Executive is aided by several additional sub-modules which are also shown in Figure 2. Once a 3-D model of the scene has been established in the viewer centered coordinate system, it is converted to an object centered representation by the "3-D Representation" module. This representation is more compact than the viewer centered description, and corresponds to a world model organized around the static components of the scene which do, in fact, dominate the scene. In the case of "roads", this is the representation passed to the Navigator Module for planning a path. This 3-D representation is also used by the "Scene Predictor" to focus attention on small areas of the visual field in which important objects are located, even after the vehicle has traveled blind; it is the foundation upon which the feed-forward mode operates. Finally, the Vision Executive can control the pointing of the camera via a "Sensor Control" module. Thus, in seeking to find a landmark or road, for example, the Executive establishes the visual field.

Three additional modules comprise our architecture as it currently stands. A "Planner Module" is responsible for establishing the overall goals of the system and assigning priorities to these goals. As we are concerned with navigation, these goals are typically location goals, either in a map, or relative to something like a landmark located in a map. For "road following" a goal may be "move to point N on the road map." It is also appropriate that the Planner be responsible for overall resource allocation as this is where the "time sensitivity" of the system resides. Hence, priorities can be established and altered when deemed necessary. The "Navigator Module" is a special purpose planning module. It is responsible for generalized path planning, as described in Section 2. The Navigator must also track the position of the vehicle through the 3-D representation as it moves blindly, using "travel" data from the Pilot. Once the Navigator establishes a particular path for some short distance, it passes the path to the "Pilot Module" which interprets this path into steering and motion commands for the vehicle. The Pilot is also responsible for monitoring the dead reckoning (and inertial navigation) unit aboard the vehicle. It should convert wheel shaft encoder readings and gyroscope headings into directional travel since the last time "travel" was requested by the Navigator.

3.3 Road Following

By accessing the computational capabilities which reside in the modules in a particular order, the system can exhibit certain behaviors. Thus, in a modular system, the "flow of control" through the system plays a central role. In Figure 2 we have indicated the flow of control between modules which supports the "road following" task. Beginning with the vehicle at a standstill, and the road somewhere in the visual field, the Planner specifies the goal to reach some point or distance down the road. This location goal is passed to the Navigator which must specify the visual attentive goal of "find the road in the field of view", passing it to the Vision Executive. The vision system must first find the road and so the bootstrap mode of processing is selected, and an image is accepted into the Image Processing Module. The Executive also selects which image processing procedure should be utilized; we currently use "linear feature extraction." Linear features which dominate the scene (are expected to reflect road boundaries and markings, and so) comprise the 2-D symbolics passed back to the Executive in the form of end-points of each feature. These symbols are then passed to the Knowledge Base where they are grouped into "pencils of lines." A road with turns or changes in slope in the field of view will give rise to several pencils in the same image. The pencils are then returned to the Executive where they are then interpreted one at a time, from bottom to top of the image (corresponding to "near to far" in the world). Each pencil is passed to the Geometry Module where it is given a 3-D shape interpretation in the form of parallel lines on a planar surface patch. This is essentially "shape from contour"; it is capable of recovering road geometry over rolling terrain (cf. Section 4). Thus, the 2-D pencils of lines are converted to 3-D parallels which are relayed to the Knowledge Base via the Executive. As each set of parallels arrives in the Knowledge Base, rule-based reasoning about road structures is used to decide if the parallels and corresponding surface patch "make sense." If not, alternate groupings of lines are considered. (A more mature system would utilize a variety of symbolic descriptions, fusing them with knowledge to interpret the scene.) Each successive pencil, interpreted as parallels, is checked for consistency with already accepted parallels, thereby building up a viewer-centered 3-D description of the road out to a distance of some 60 meters or so. Parts of the road are labeled as "boundaries, center line, lane markers or shoulder boundaries." That is, the object "road" is recognized via its attributes in the world, particularly their spatial relations. This scene model is then passed to the 3-D Representation module which converts the viewer centered model of the road (consisting of a sequence of parallel lines in space relative to the camera) into a compact, object centered representation in which each planar patch is described relative to the previous patch, and the road width and number of lanes is specified. The vehicle is then located relative to the first planar patch. Before passing this representation on to the Navigator (the vehicle still has not moved), the Executive uses the Scene Predictor to select small windows ($5^o \times 5^o$) near the bottom of the visual field inside of which the left and right road boundaries are located. The Executive then utilizes the local image processing of the feed-forward mode to refine the locations of the road boundaries. Once the road boundaries have been established, the system tries to continue to track them through the image domain, checking their 3-D interpretations for consistency with the rules. Once the road is extracted from the first image, and its boundaries have been refined giving rise to a more accurate 3-D representation, this representation is passed on to the Navigator. (This road map can be pasted on to a map of a previous segment of road and passed to the Planner for future use.) The Navigator then plans a path down the road and passes segments of this path to the Pilot. The Pilot decomposes the path into motion commands and the vehicle begins to move. Periodically, the Pilot returns the distance and heading actually travelled to the Navigator. Every 10 meters, the Navigator informs the Vision Executive of position and the Executive requests that a new image be taken. The Executive also passes this position to the Scene Predictor which, utilizing the previous 3-D representation, determines the appearance of the road boundaries from this new vantage point and selects new windows near the bottom of the new image inside of which the road boundaries are to be found. The new image is then processed in the feed-forward mode and the system continues on in this way until instructed to stop. In this manner, the vehicle travels over road segments about 10 meters long using a map it created from an image taken some 10 meters ahead of that segment, maintaining a continuous motion. Utilizing a VAX in the feed-forward mode, the system can support road following at about 3 km/hr. Using a VICOM image processor, this speed can be doubled [8].

We should point out that the size of $5^\circ \times 5^\circ$ used for focusing attention in the feed-forward mode reflects the uncertainty in the position and heading of the vehicle as determined by the dead reckoning system (since the last image was taken about 10 meters earlier), as well as uncertainty in camera pointing. Mainly, it must accommodate uncertainties in the visually constructed road map. This window was selected on the basis of a positional error of about .3 meters and a pointing error of 1° with respect to the road boundaries [1].

There are many other capabilities we must still develop to support general road following. For example, our system cannot make it through an intersection in the feed-forward mode, nor does it do any obstacle avoidance. Landmark recognition, global positioning and route planning on a network should also be incorporated.

4. COMPUTATIONAL CAPABILITIES OF MODULES

Having described above the nature of the responsibilities assigned to the individual modules, as well as how they are linked by a flow of control to support the task of road following, we now turn to the specific computations that take place within the modules. The nature of these computations are suggested by the points listed within the module boxes of Figure 2. Some of the capabilities shown in the diagram are not yet installed in our system, though they are suggested by the modular decomposition itself and their relation to currently existing capabilities. We consider each module separately.

4.1 IMAGE PROCESSING

We have developed algorithms for the extraction of dominant linear features from entire (gray level) images as well as gray and color segmentation routines. These analyses provide independent representations of the information contained in the images in the form of *boundary based* and *region based* descriptions, respectively. These routines are relevant to the bootstrap mode of processing described earlier. Related versions were also developed to support the feed-forward mode. Examples of this processing on a road image are shown in Figures 3 and 4. A detailed description of the image processing steps is presented in [15]. A host of other low-level vision capabilities can be installed in this module such as local edge and zero-crossing detectors, image change detectors and multiresolution filters. The capabilities we have already installed combine low-level operations with certain grouping procedures to derive particular symbolic descriptors.

Thus, our linear feature detector relies on the grouping of pixels with similar gradient orientation, yielding a linear feature with global support (and poor localization) in the bootstrap mode and one with local support (and good localization) in the feed-forward mode. The segmentation procedures are based on edge-preserving smoothing followed by a connected-components analysis. The color version of the segmentation procedure is far less sensitive to parameters than is the gray version. (It also provides the additional measure of "color" for each segment.) Linear feature extraction is completely automatic. Since the knowledge-based reasoning module in our system cannot currently fuse multiple sources of data (such as both linear features and regions), we utilize only the linear features when running the entire system.

4.2 VISUAL KNOWLEDGE BASE

This module implements rule-based reasoning fc. two separate vision tasks (and so, perhaps, should have been constructed as two different modules); seeking significant groupings of symbols derived from an image and checking consistency of 3-D shape recovery with models of objects (e.g., roads). Many types of symbolic groupings can be considered in general [14], though for purposes of road following we concentrated on the grouping of linear features into pencils of lines.

Pencils are determined by spatial clustering of intersections between pairs of lines in order to suggest a vanishing point. A sequence of vanishing points are adopted when grouping lines from the bottom of an image to the top (corresponding to "near to far" in the world). In the context of road following, these groupings represent hypotheses about road boundaries and markings, and the road geometry itself when more than one grouping is found (for example turns and changes in ground slope). The bootstrap results of Figure 3 would be grouped into two pencils (the left road boundary line would be split in two) roughly delineating the turn in the road; the horizon lines would be discarded as not consistent with "converging pencils." The feed-forward results in Figure 4 would yield a sequence of many pencils (as the road boundaries are tracked from their predicted locations at the bottom of the image to the top). These pencils more accurately reflect the road geometry and structure of the terrain. We shall soon install a region grouping capability, designed to operate on the segmentation results, such that the global boundaries of grouped regions form a pencil. This is relevant to recognition of patchy road surfaces.

We have not as yet installed "landmark recognition" in our system, though the Hough-based edge clustering approach described in [12] would belong in this module as well. This is an example of recognition from image domain groupings directly, without first attempting 3-D shape recovery.

Once pencils have been grouped and assigned a 3-D interpretation by the Geometry Module (see below), the knowledge-based module attempts to reason about the consistency of the successive surface patches that comprise the hypothetical road. Changes in surface slope and 3-D symmetries of the road are typical attributes that are considered. If the surface patches and corresponding road segments satisfy the constraints, the interpretation of a "road and its parts" are assigned and associated with a scene model. Further details about the reasoning approach will be described in [16].

A major extension to this reasoning process that must be considered is the ability to combine the independent symbolic descriptions extracted by the Image Processing module. The complementary nature of the boundary-based and region-based descriptions illustrated in the lower right quadrant of Figure 3 is clear. For example, grouping of lines into pencils can be used to focus attention on the segments bounded by the pencil. In fact, the consistency of the pencil with the grouped segments can be used to select parameters for the segmentation process (which can be parameter sensitive). When agreement is found, the segmentation results can be used to construct a model of the road out to a much greater distance (some 100 meters). This kind of information can be useful for route planning. Consistency between complementary descriptions also lends greater confidence to the labeling of the scene. The challenge of combining independent evidence is a well recognized and unsolved problem in Computer Vision.

4.3 GEOMETRY

The Geometry Module converts the grouped symbolics in the image domain to a viewer centered 3-D description of objects in the scene. A variety of "shape from" techniques have been suggested for accomplishing this in general (as listed in the diagram), leading to a representation termed the "2.5-D sketch" by Marr [14]. When several methods are employed, their results must be combined in a kind of "integrated 2.5-D sketch." Currently, our system utilizes a single method for recovering shape from monocular imagery; it is essentially "shape from contour." We will soon be adding a "stereo" capability as well.

Our method of shape recovery is really model driven. We can invert the perspective transformation of the imaging process if we adopt the following three (road) model-based assumptions:

1. Pencils in the image domain correspond to planar parallels in the world.
2. Continuity in the image domain implies continuity in the world.
3. The camera sits above the first visible ground plane (at the bottom of the image).

Our 3-D reconstruction will then characterize a road of constant width, with turns and changes in slope and bank. The reconstruction process amounts to an integration out from beneath the vehicle into the distance, along local parallels over topography modeled as a sequence of planar surface patches.

An example of the method is shown in Figures 5 and 6. Figure 5 illustrates a set of linear features that are imagined to be extracted from a sequence of four images taken as the vehicle moves (the corresponding positions are labeled "a, b, c, d" in Fig. 6). The linear features are grouped into pencils according to their vanishing points. The pencils are numbered to provide continuity from image to image (and will be interpreted as the numbered surface patches in Fig. 6). Each pencil will be interpreted as parallel lines on a planar surface patch, and so a reconstruction of the road will resemble that shown in Figure 6. The points in the image where line segments actually meet are called "continuity points", marked "c" in Fig. 5a. The point marked "v" in Fig. 5a illustrates a "vanishing point" determined by the intersection of linear features. The points where the first pencil of lines leave the bottom of the visual field in the first image (taken when the vehicle starts) are termed "initialization points," marked "in" in Fig. 5a. We adopt a viewer centered coordinate system in which the origin is at the center of the camera lens and the Z-axis points along the center of the field of view. The X-axis is chosen horizontal, pointing to the left of the viewer. The Y-axis would then be vertical if the camera were not tilted. We consider image coordinates scaled to a focal length of unity; hence,

$$x = X/Z \quad \text{and} \quad y = Y/Z \ . \tag{1}$$

The i^{th} planar surface patch is then described in the 3-D coordinates by

$$A^i X + B^i Y + C^i Z = 1 \ , \tag{2a}$$

and so in image coordinates by

$$A^i x + B^i y + C^i = 1/Z \ . \tag{2b}$$

It is assumed that the camera sits at a known height H above the ground and has a known tilt τ with respect to the horizontal. Then, regardless of the pan angle (or orientation of the vehicle with respect to the road), the coefficients of the first ground plane are determined by Assumption 3 above,

$$A^1 \equiv 0 \ , \quad B^1 = -\frac{\cos\tau}{H} \ , \quad C^1 = \frac{\sin\tau}{H} \ . \tag{3a}$$

The depth to the initialization points can then be determined by equation (2b) as

$$\frac{1}{Z_{in}} = \frac{1}{H}\left(\sin\tau - y_{in}\ \cos\tau\right) . \tag{3b}$$

To determine the geometry of the road in the distance, it is necessary to solve for the coefficients of each successive planar patch, and then obtain depth to points on the road boundaries using equation (2b). The coefficients are determined by three conditions. Two conditions are provided by the assumed continuity of the left and right road boundaries in the world (when they are continuous in the image). Starting from the first planar patch, we can easily determine depth to the first continuity points at the end of the first planar patch. We then require that depth to these points, based on the parameters of the second planar patch, be the same, as these points reside at the boundary between the first and second planar patches. Thus, each side of the road generates one continuity condition. This is true in general between any successive planar patches. We express these conditions in the form

$$A^i x_{c_l} + B^i y_{c_l} + C^i = 1/Z_{c_l}^{i-1} \ , \tag{4a}$$

$$A^i x_{c_r} + B^i y_{c_r} + C^i = 1/Z_{c_r}^{i-1} \ . \tag{4b}$$

The third condition expresses the fact that the vanishing point of a pencil represents the projection of the point at infinity (i.e., $Z \rightarrow \infty$):

$$A^i x_{v_i} + B^i y_{v_i} + C^i = 0 \ . \tag{4c}$$

Equations (4a,b,c) form a set of three linear equations in the three unknown coefficients describing the i^{th} planar patch. Thus, starting from the first ground plane, one works outward from the vehicle, building a model of the road topography in the viewer centered coordinate system.

Our current Geometry Module implements this method of reconstructing road topography. As each planar patch is recovered, the linear features are subdivided into segments which are physically opposite each other on the road in 3-D. This simplifies the symmetry checks imposed by the Visual Knowledge Base following shape recovery, and is also required for the Predictor. It should be clear that this method of shape recovery is tightly coupled to the hypothesis that the pencils of lines correspond to road boundaries and markings. The hypothesis itself, however, is generated following the grouping operation. That is, the "shape from" process is being imposed with a particular generic object in mind.

4.4 VISION EXECUTIVE

This module does not really perform any computations as yet. It maintains the flow of control through the system in order to exhibit a behavior, such as road following. As the contents of the Knowledge Base are expanded to incorporate other object models, the Vision Executive will have to make decisions about which packets of knowledge to activate within the Knowledge Base, and in what order. That is, it will have to construct a "knowledge strategy" in the form of "meta-rules."

4.5 3-D REPRESENTATION

This module converts the 3-D viewer centered representation of the road scene into an object centered description. The description is in the form of a file which lists a set of road attributes at each "roadmark" set down. Roadmarks are placed (by this module) along the centerline of the reconstructed road model, at the beginning of each new planar patch. Roadmarks will also be placed at intersections (once our system learns to recognize them). Figure 7 illustrates this representation in terms of roadmarks. For each roadmark, the file contains the distance from the previous roadmark, and the Euler angles which describe the three (non-commuting) angles of rigid body rotation of a local coordinate system tied to the road (described in more detail below). Thus, each segment of road is described relative to the previous segment of road. This is important for several reasons. As the vehicle moves, it "explores the object called Road," experiencing changes in road geometry between segments of road. The absolute geometry of a road segment is irrelevant (and geometry relative to one particular view is even more meaningless). Moreover, errors in road geometry due to the visual process will be confined to local errors, rather than being compounded over the entire distance traveled. Similar views have been described by Brooks [17]. In addition to the distance and Euler angles relative to the previous roadmark, the road width, number of lanes, and presence of a shoulder are encoded in the 3-D representation. Finally, the location of obstacles and the vehicle itself, relative to the nearest roadmarks, are included in the representation created.

The conversion from viewer centered coordinates to the object centered (Euler angles) frame is illustrated in Figure 8. Here, two planar patches are represented by local coordinate frames attached to the road at points A and B. The unit vectors $\boldsymbol{n}$, $\boldsymbol{l}$ and $\boldsymbol{t}$, aligned with the local normal, longitudinal and transverse directions at each of these points are easily obtained from the viewer centered results provided by the Geometry Module. The two local systems are related by a rigid body rotation, specified by three Euler angles θ, ϕ and ψ which correspond to the three rotations shown in the figure. As such rotations are finite, their order of application is important. The sequence of transformations is shown in Figure 8; first: a rotation about the normal axis $\boldsymbol{n}$ by the *turn angle* θ;

second: a rotation about the new transverse axis $\boldsymbol{t}$ by the *slope angle* ϕ; third: a rotation about the final longitudinal axis $\boldsymbol{l}$ by the *bank angle* ψ. Each rotation is a two-dimensional rotation about a different axis; they are given by "turn, slope and bank" matrices $\boldsymbol{T}(\theta)$, $\boldsymbol{S}(\phi)$, $\boldsymbol{B}(\psi)$:

$$\boldsymbol{T}(\theta) = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{5a}$$

$$\boldsymbol{S}(\phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{pmatrix} \tag{5b}$$

$$\boldsymbol{B}(\psi) = \begin{pmatrix} \cos\psi & 0 & \sin\psi \\ 0 & 1 & 0 \\ -\sin\psi & 0 & \cos\psi \end{pmatrix} \tag{5c}$$

The net rigid rotation is the product of these three matrices $\boldsymbol{R}(\theta, \phi, \psi) = \boldsymbol{B}(\psi)\, \boldsymbol{S}(\phi)\, \boldsymbol{T}(\theta)$, yielding

$$\boldsymbol{R} = \begin{pmatrix} \cos\theta\cos\psi + \sin\theta\sin\phi\sin\psi & \sin\theta\cos\psi - \cos\theta\sin\phi\sin\psi & \cos\phi\sin\psi \\ -\sin\theta\cos\phi & \cos\theta\cos\phi & \sin\phi \\ -\cos\theta\sin\psi + \sin\theta\sin\phi\cos\psi & -\sin\theta\sin\psi - \cos\theta\sin\phi\cos\psi & \cos\phi\cos\psi \end{pmatrix}. \tag{6}$$

The Euler angles θ, ϕ and ψ used in the representation can be determined by noting that the elements of the rotation matrix $\boldsymbol{R}(\theta, \phi, \psi)$ may be identified with the *direction cosines* relating the two local reference frames at points A and B. Hence, each element of $\boldsymbol{R}$ can be assigned a numerical value obtained from the 3-D data in the viewer centered reference frame. We need only compute the values of three elements and bound the rotation angles by $\pm 90°$. In this manner we find

$$\sin\phi = \boldsymbol{l}_B \cdot \boldsymbol{n}_A \ , \tag{7a}$$

$$\sin\theta \cos\phi = -\boldsymbol{l}_B \cdot \boldsymbol{t}_A \ , \tag{7b}$$

$$\sin\psi \cos\phi = \boldsymbol{t}_B \cdot \boldsymbol{n}_A \ . \tag{7c}$$

A complete discussion of Euler angles and direction cosines may be found in [18].

4.6 PREDICTOR

This module is used to focus the attention of the system on a small part of the visual field. Windows in the field of view are determined inside of which the near part of the road boundaries are located, following the vehicle's blind travel through the 3-D representation created from an earlier image. This is accomplished by essentially transforming the 3-D data used to create the most recent representation according to a rigid body translation and rotation associated with the vehicle's motion (as is familiar in Computer Graphics). The components of travel used in constructing this transformation are obtained from the dead reckoning (and inertial navigation) system aboard the vehicle. Once the 3-D representation is transformed accordingly, we solve for the intersection of the road boundaries with the periphery of the field of view. Windows of $5^o \times 5^o$ are then placed over these points at the lower part of the visual field.

As additional capabilities are added to our system, new prediction routines must be created in order to support these capabilities in the feed-forward mode. For example, predicting the apparent locations of stationary obstacles will be very similar to the method already developed. Predicting locations of moving objects, however, will require predictions of their trajectories through the world during the blind travel time.

4.7 SENSOR CONTROL

This module is essentially an interface to the device driver associated with the pan and tilt mechanism of the camera. Computations consist of conversions between relative and absolute pointing angles.

4.8 NAVIGATOR

This module must provide computational capabilities in support of generalized path planning, as described in Section 2. Thus, it must be able to establish vehicle position from known landmarks sighted, generate region graphs from visibility/traversibility data, generate region sequences between current position and goal, and plan paths down corridors of free space while avoiding obstacles. Our current Navigator is specific to following obstacle free roads (corridors) and so path planning consists of smoothly changing the heading of the vehicle in accordance with the 3-D representation derived from the visual process. This is accomplished by computing cubic arcs as asymptotic paths. Given the current position and orientation of the vehicle, a cubic arc is derived which begins at the vehicle, slopes in the direction of vehicle heading, and terminates 13 meters along the road centerline, in the direction of the centerline. This cubic arc is used as a path for the next 3 meters (about one vehicle length) at which point a new cubic arc is derived which terminates another 13 meters ahead. That is, the vehicle is always steered towards a point which is 13 meters ahead of it, along the centerline. Thus, the vehicle's path will be asymptotic to the road centerline. (If the road has more than one lane, we displace the centerline to the middle of a lane.) This yields paths with smooth changes in heading. When obstacles are introduced onto the road, we must adjust the termination point of a cubic arc so that no part of the arc intersects an obstacle nor crosses the road boundary.

In addition to planning spatial trajectories, the Navigator must plan temporal ones as well, essentially adjusting the speed of the vehicle in accordance with the upcoming demands of the road geometry. As our vision system generates a model of the road to quite far ahead of the vehicle, the Navigator should be able to perform simple rule-based reasoning about speed. Thus, the vehicle could speed up on straight-aways, and slow down on turns. We currently operate at a single vehicle speed.

In order to support the focus of attention mechanism in the feed-forward mode, the Navigator must track vehicle position through the 3-D representation in response to dead reckoning data provided by the Pilot. This information is made available to the Vision Executive at its request. This is currently implemented in our system.

4.9 PILOT

This module converts the cubic arcs obtained from the Navigator into a sequence of conventional steering commands used over the next 3 meters. They decompose a curved path into a set of short, straight line segments. These motion commands must then interface to the motor controls of the vehicle. (In our case, the interface is to the motion control software of a robot arm, as described in the next section.) The Pilot is also responsible for sending dead reckoning data from the vehicle to the Navigator. The Pilot converts the raw data into measured travel, and returns this information to the Navigator several times per second.

4.10 PLANNER

Our current Planner is quite simple; it merely specifies a "distance goal", e.g. move to the point 60 meters further down the road. A mature planning module could have arbitrary complexity, specifying high level navigation goals, assigning priorities to these goals, monitoring progress as a function of time and constructing contingency plans. It would also be responsible for allocating computational resources throughout the system. However, until the vehicle can exhibit a variety of behaviors, it seems rather premature to concentrate on issues of high level planning.

5. PHYSICAL SIMULATION

We have implemented our system as a set of concurrent, communicating processes running on VAX 11/785 under Berkeley UNIX 4.2. In order to test the system as a whole, in the absence of a real outdoor vehicle, we have constructed a physical simulation apparatus as shown in Figure 9a. We constructed a road network to a scale of 1:96 (i.e. 8 feet to the inch) on a wooden terrain board. In place of a vehicle we use a small solid state CCD black & white camera, carried by an American Robot MERLIN robot arm. The center of the camera lens is mounted beneath the last axis of the robot arm in order to decouple the translational and rotational motions of the "vehicle" as much as possible. As the "vehicle" navigates (with three degrees of freedom) over a two-dimensional surface, while the robot arm navigates (with six degrees of freedom) through a three-dimensional world, it was necessary to modify the apparatus in order to model a vehicle on the ground. Figure 9b shows a close-up view of the camera with three linear position encoders mounted on it. These sensors (spring mounted variable resistors) are dragged over the terrain board as the camera moves. By monitoring them, we can determine the height and orientation of the camera with respect to the terrain below it. Thus, in order to model a vehicle on the ground, we servo off of these three encoders, adjusting camera position and orientation to maintain a fixed height and tilt of the camera above the board. The navigation commands for the robot arm are thus derived from the vehicle's Pilot Module along with the servo control from the three linear encoders. These navigation commands provide input to the motion control software that resides on the arm's own computers. As the motion of the robot arm is known rather precisely, we can simulate the return of a dead reckoning system quite simply, adding a small perturbation to the data in order to simulate drift or wheel slippage.

In general, we have found this simulation to be quite adequate for purposes of developing and implementing our visual navigation system. In order to simulate the active ranging capabilities of a laser scanner that may be mounted upon a real vehicle, we plan to add a "structured light" component to our own system. This will be used in the near future for purposes of avoiding nearby obstacles.

CONCLUDING REMARKS

We have developed our first generation visual navigation system capable of navigating on roadways at about 3 km/hr. Motivated by this application, a more general, modular system architecture was created in which "computational capabilities" and "flow of control" play distinct roles. Our system runs as a set of concurrent processes communicating with each other in a manner that resembles distributed computing. Thus, it will be rather easy to map our system onto such multiprocessors. The addition of a dedicated image processing engine, running in parallel with the rest of the system on a VAX (or possibly a Butterfly) will provide us with a real opportunity to study the issues of timing and achievable speed on a distributed parallel system.

Many extensions are planned for our system, such as stereo vision in the Geometry Module, reasoning about multiple sources of data (linear features and segmented regions) in the Visual Knowledge Base, and planning paths around obstacles and through road networks in the Navigator Module. Future enhancements to the Image Processing Module are expected as well. New capabilities such as positioning via landmarks and obstacle avoidance will greatly expand our system's performance.

It is a pleasure to acknowledge the assistance of John Bosque, Roger Eastman, David Harwood, Babu Srinivasan and Mark Westling at various times throughout the development of this navigation system.

REFERENCES

[1] A.M. Waxman, J. Le Moigne and B. Srinivasan, "Visual Navigation of Roadways", 1985 IEEE Int. Conf. Robotics and Automation, pp. 862-867, St. Louis, MO, March 1985.

[2] H.P. Moravec, "Robot Rover Visual Navigation", Ann Arbor, MI: UMI Press, 1981.

[3] R. Chatila, "Path Planning and Environmental Learning in a Mobile Robot System", Proc. ECAI, Orsay, August 1982.

[4] R. Chavez and A. Meystel, "Structure of Intelligence for an Autonomous Vehicle", 1984 IEEE Int. Conf. Robotics, pp. 584-591, Atlanta, GA, March 1984.

[5] C.E. Thorpe, "FIDO: Vision and Navigation for a Robot Rover", Ph.D. thesis, Dept. Computer Science, Carnegie-Mellon University, December 1984.

[6] R.A. Brooks, "A Mobile Robot Project", MIT AI Working Paper 265, February 1985.

[7] J.L. Crowley, "Navigation for an Intelligent Mobile Robot", IEEE J. Robotics and Automation, vol. 1, pp. 31-41, 1985.

[8] L.S. Davis and T. Kushner, "Road Boundary Detection for Autonomous Vehicle Navigation", University of Maryland, Center for Automation Research Tech. Report 140, July 1985.

[9] D.T. Kuan, J.C. Zamiska and R.A. Brooks, "Natural Decomposition of Free Space for Path Planning", 1985 IEEE Int. Conf. Robotics and Automation, pp. 168-173, St. Louis, MO, March 1985.

[10] E. Koch, C. Yeh, G. Hillel, A. Meystel and C. Isik, "Simulation of Path Planning for a System with Vision and Map Updating", 1985 IEEE Int. Conf. Robotics and Automation, pp. 146-160, St. Louis, MO, March 1985.

[11] R. Chatila and J.-P. Laumond, "Position Referencing and Consistent World Modeling for Mobile Robots", 1985 IEEE Int. Conf. Robotics and Automation, pp. 138-145, St. Louis, MO, March 1985.

[12] F.P. Andresen and L.S. Davis, "Visual Position Determination for Autonomous Vehicle Navigation", University of Maryland, Center for Automation Research Tech. Report 100, November 1984.

[13] S. Kambhampati and L.S. Davis, "Multiresolution Path Planning for Mobile Robots", University of Maryland, Center for Automation Research Tech. Report 127, May 1985.

[14] D. Marr, "Vision: A Computational Investigation into the Human Representation and Processing of Visual Information", San Francisco, CA: Freeman, 1982.

[15] J. Le Moigne, A.M. Waxman, B. Srinivasan and M. Pietikainen, "Image Processing for Visual Navigation of Roadways", University of Maryland, Center for Automation Research Tech. Report 138, July 1985.

[16] J. Le Moigne, "Domain-dependent Reasoning for Visual Navigation of Roadways", University of Maryland, Center for Automation Research Tech. Report in preparation.

[17] R.A. Brooks, "Visual Map Making for a Mobile Robot", 1985 IEEE Int. Conf. Robotics and Automation, pp. 824-829, St. Louis, MO, March 1985.

[18] H. Goldstein, "Classical Mechanics", Reading, MA: Addison-Wesley, 1959.

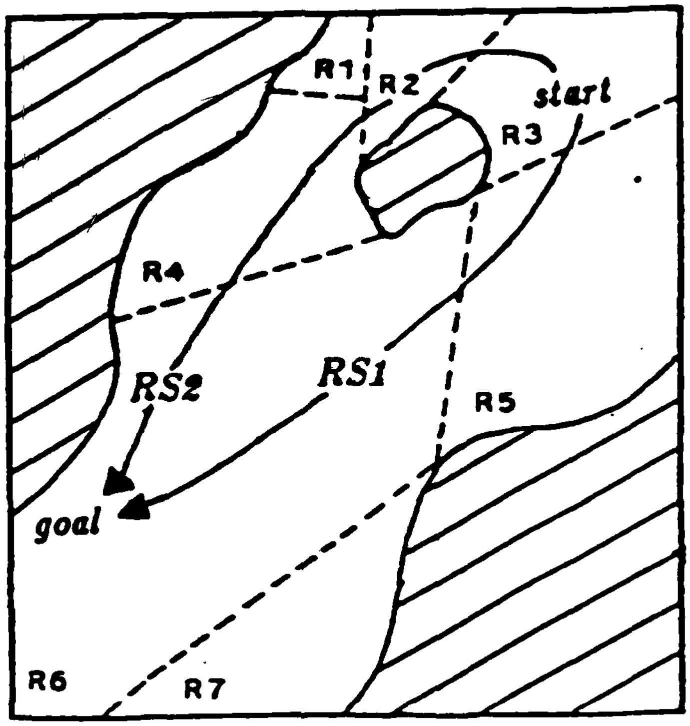


Fig. 1 - Long Range Navigation database resembles a map, segmented into regions of uniform traversibility and visibility of landmarks. On this scale, paths resemble region sequences such as RS1 = (R3, R5, R6) and RS2 = (R3, R2, R4, R6).

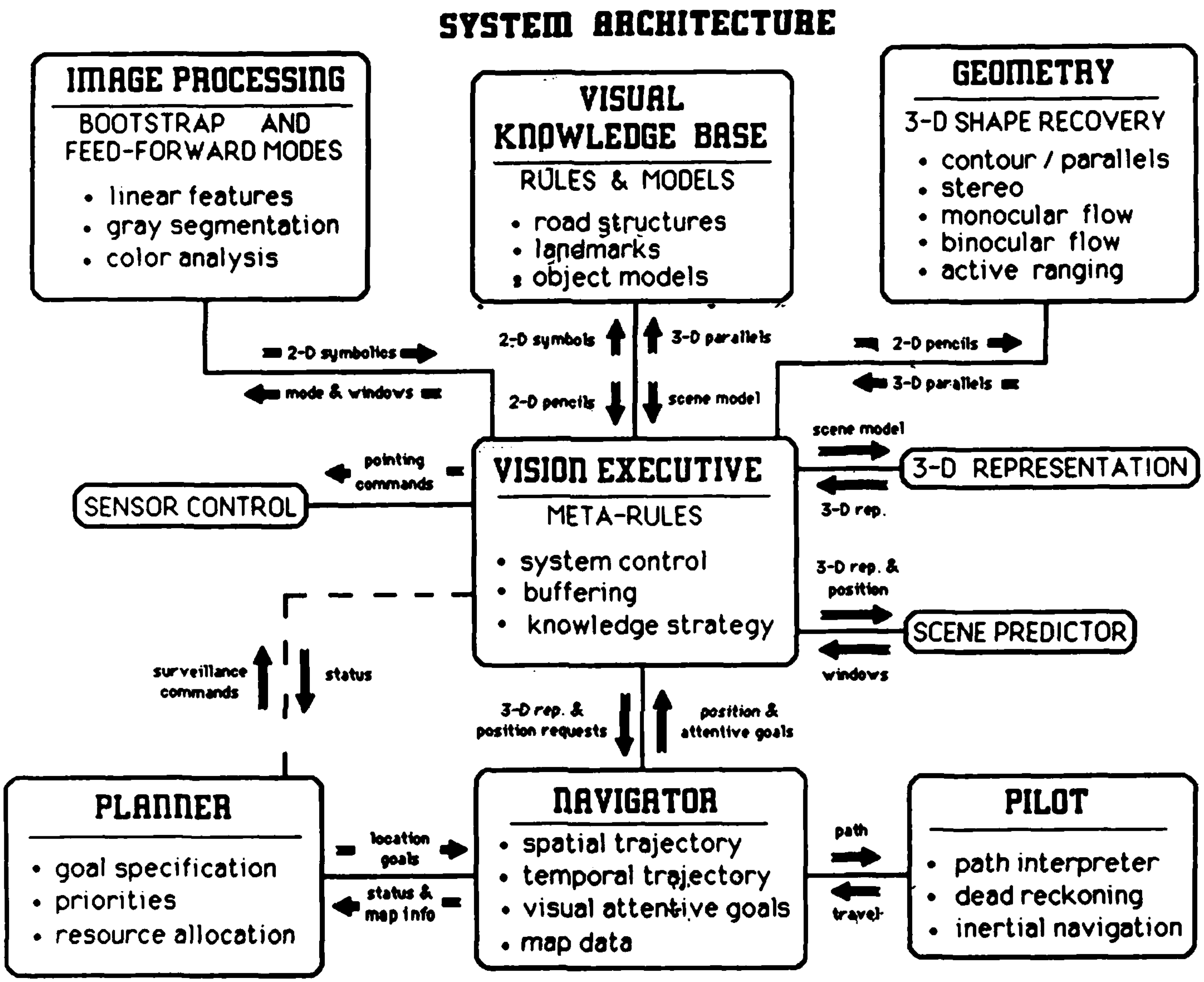


Fig. 2 - Modular system architecture for visual navigation. The flow of control and data over the interfaces supports road following.

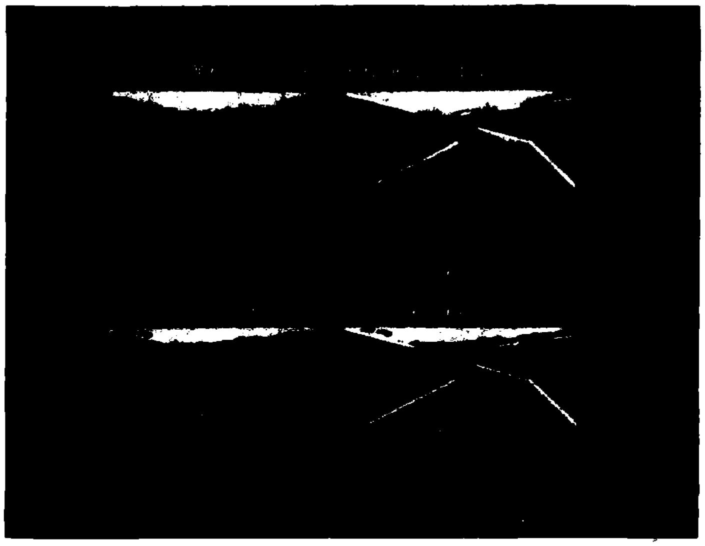


Fig. 3 - Image processing in the bootstrap mode. Dominant linear features and gray segmentation provide symbolic descriptions. Processing of the entire image is required.

Fig. 4 - Linear feature extraction in the feed-forward mode. Processing is confined to the small windows. Lower windows are placed by a Predictor, road boundaries are detected and tracked.

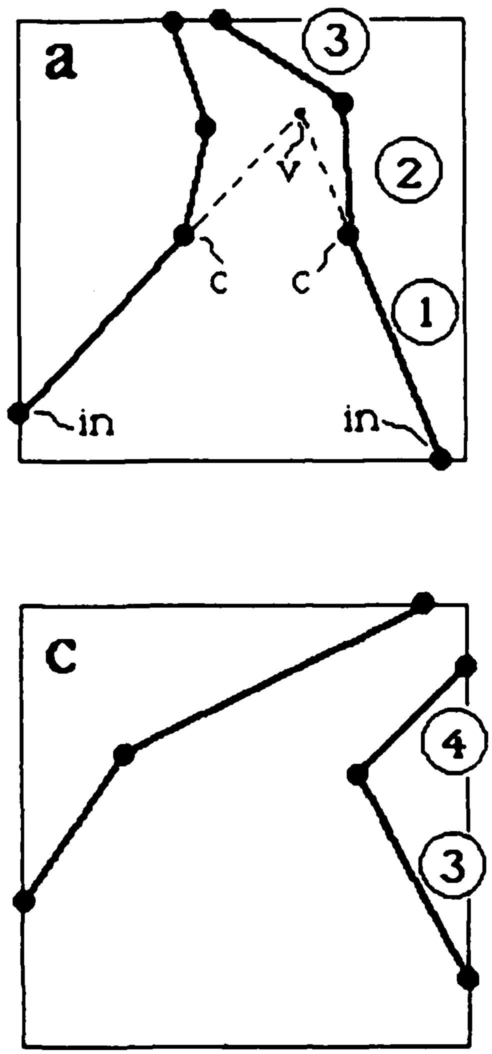


Fig. 5 - A (simulated) sequence of linear features detected from four positions as the vehicle moves down a road. The lines are grouped into pencils for interpreting 3-D shape.

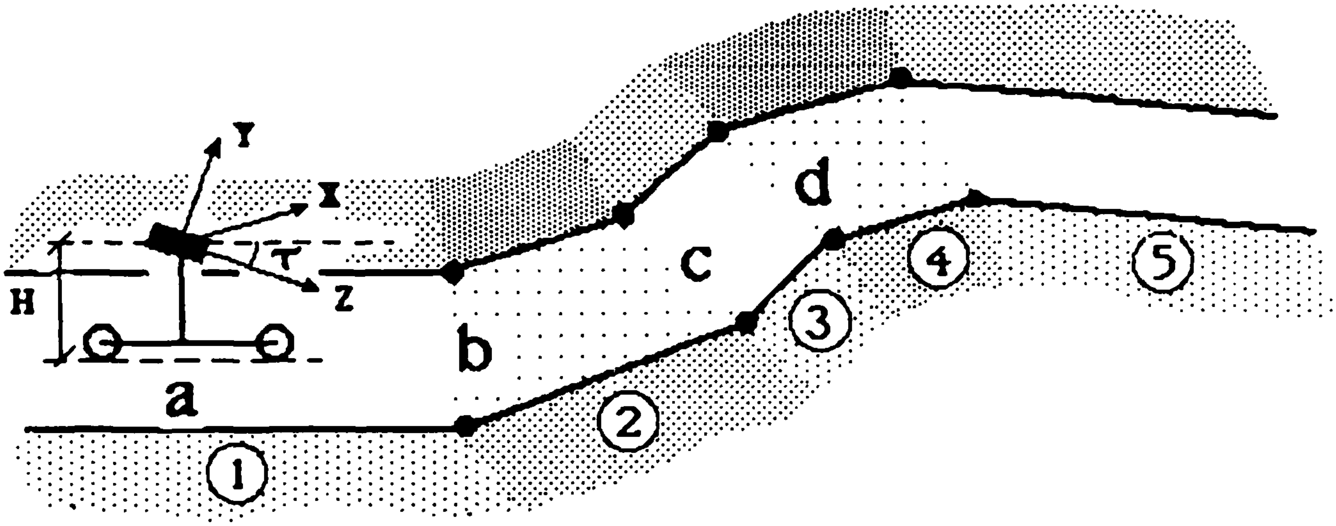


Fig. 6 - 3-D reconstruction of road geometry over terrain obtained from pencils in Fig. 5. Road is approximated as a sequence of parallel line segments on planar surface patches.

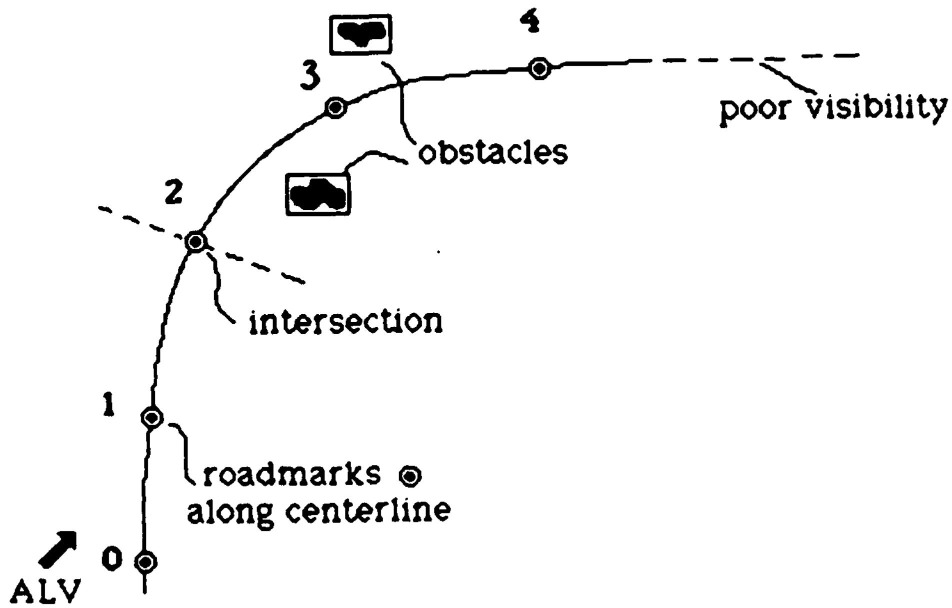


Fig. 7 - 3-D representation of road in object centered reference frame. Roadmarks correspond to data fields in a file. Each field describes a road segment relative to the previous segment. The vehicle (ALV) and obstacles are located relative to the road.

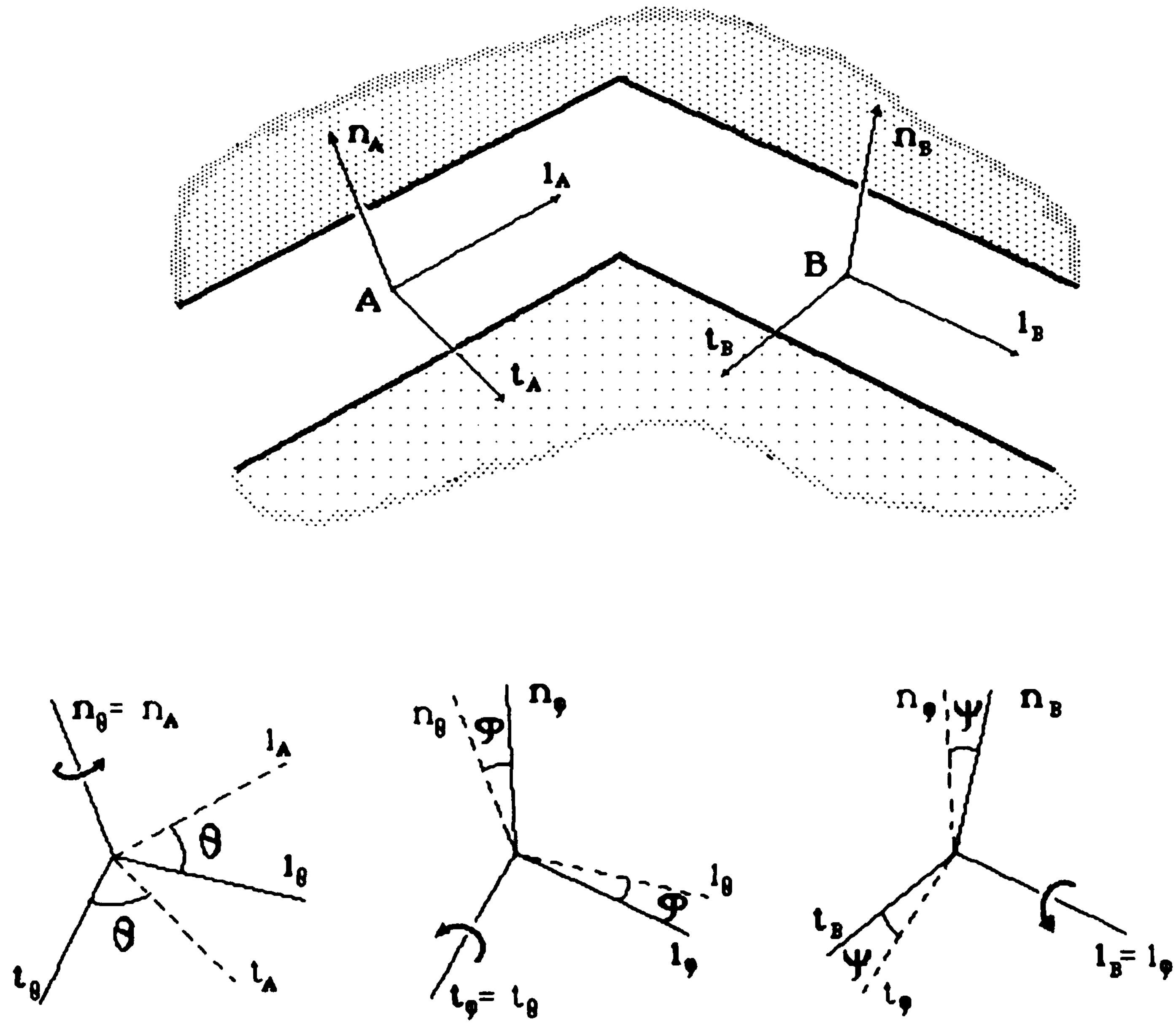


Fig. 8 - Two segments of a reconstructed road and their local reference frames A and B. These frames are related by a translation along the surface and a rigid body rotation. The sequence of rotations corresponding to "turn, slope and bank" are illustrated along with the Euler angles.

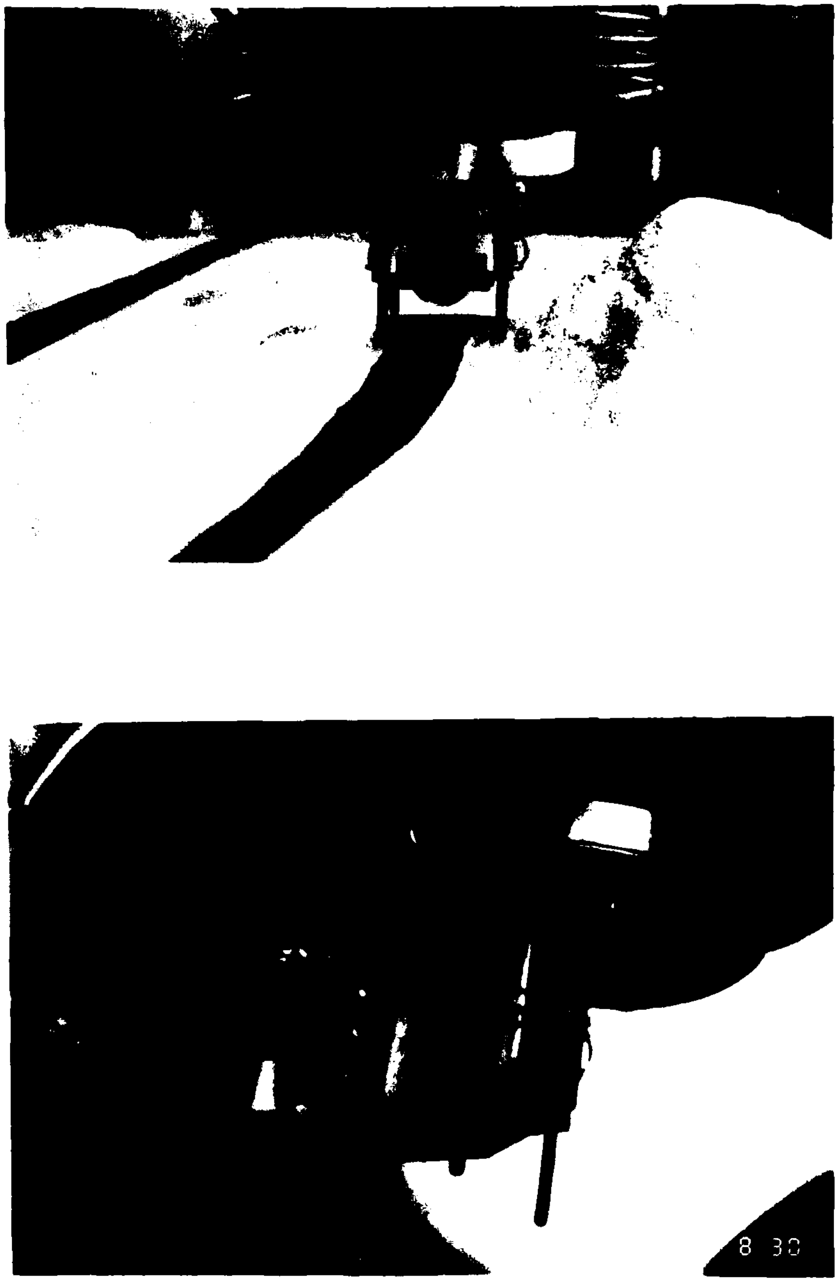


Fig. 9 - ALV simulation apparatus: (a) a mini-CCD camera is carried over a terrain board by a robot arm, the road network is scaled by 96:1; (b) three linear position encoders are mounted to the camera and touch the board, they are used to adjust camera height and orientation.

END

FILMED

12-85

DTIC